UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/288,263	04/08/1999	HIROYUKI WAKI	NAK1-BG55	7236

21611 7590 12/29/2003

SNELL & WILMER LLP
1920 MAIN STREET
SUITE 1200
IRVINE, CA 92614-7230

EXAMINER	
LAFORGIA, CHRISTIAN A	
ART UNIT	**PAPER NUMBER**
2131	19

DATE MAILED: 12/29/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/288,263	WAKI ET AL.	
	Examiner	Art Unit	
	Christian La Forgia	2131	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 October 2003*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *30,39-44,52 and 53* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *30,39-44,52 and 53* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____.
4) ☐ Interview Summary (PTO-413) Paper No(s). _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

DETAILED ACTION

1. The amendment filed on 16 October 2003 is noted and made of record.

2. Claims 30, 39 through 44, 52, and 53 are presented for examination.

Response to Arguments

3. Applicant's arguments filed 16 October 2003 have been fully considered but they are not persuasive.

4. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., bypassing the step of compiling the basic blocks into native code and the notifying the virtual machine of the virtual machine code blocks) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

5. With regards to the Applicant's argument that Adl-Tabatabai does not teach the basic blocks being compiled to be executed by the virtual machine, the Examiner respectfully disagrees. As taught by Adl-Tabatabai in column 2, lines 49 to 57, the Java class file consists of byte codes. According to the **Microsoft Computer Dictionary, 5th Edition**, byte codes is defined as:

> An encoding of a computer program that a compiler produces when the original source code is processed.

Still further in column 3, lines 5 and 6, we learn that the Java class files are implemented on a client computer system by a Java Virtual Machine. Additionally in column 3, lines 18 through 22, Adl-Tabatabai explicitly states that the Java Virtual Machine interprets the byte codes to execute the Java application. Therefore, Adl-Tabatabai teaches basic blocks being compiled to

be executed by the virtual machines, as basic blocks are inherent to any program with control flow.

6. See further rejections that follow.

Claim Rejections - 35 USC § 102

7. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

8. Claims 30, 39, 52 and 53 are rejected under 35 U.S.C. 102(e) as being anticipated by United States Patent No. 6,170,083 to Adl-Tabatabai, (hereinafter Adl-Tabatabai).

9. As per claims 30 and 39, Adl-Tabatabai teaches a data storage method that stores a virtual machine instruction sequence generated by compiler to be executed by a virtual machine, the data storage method comprising:

dividing the virtual machine instruction sequence into basic blocks each corresponding to an instruction block (column 5, lines 36-55);

transmitting the instruction block to the virtual machine (column 3, lines 12-18);

storing the instruction block in the virtual machine (column 3, lines 19-35);

formatting the instruction block to include:

an identifier area for storing an identifier that specifies a start position of the instruction block (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9);

a non-branch instruction area for storing non-branch instructions belonging to the corresponding basic block (column 5, lines 36-55);

a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block (column 5, lines 36-55); and

each branch instruction stored in the branch instruction area designating a branch destination using an identifier stored in one of the identifier areas (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9).

This is based on the definition of a basic block as offered by United States Patent No. 6,044,222 to Simons et al., in which a basic block is defined as

> A basic block is a contiguous sequence of instructions such that control flow can only enter at the first instruction and leave at the last instruction. Basic blocks have the characteristic that if one of the instructions is ever executed by the program, all the instructions in the basic block will be executed (assuming that no exceptions occur). A basic block typically starts at a branch label and is typically terminated by some sort of branch instruction. *Column 2, lines 15-23*

For a better graphical representation of a basic block please refer to United States Patent No. 5,923,883 to Tanaka et al., Figures 1, 2, 3, 4, 5, 6, 10, 12, 15, 16, 17, 18, 19, 20, 21, 23a, 23b, 24c, 26a, and 28b.

10. As per claims 52 and 53, Adl-Tabatabai teaches a method of storing a virtual machine instruction sequence generated by compiler to be executed by a virtual machine, the method comprising:

dividing the virtual machine instruction sequence into basic blocks each corresponding to an instruction block (column 5, lines 36-55);

transmitting the instruction block to the virtual machine (column 3, lines 12-18);

storing the instruction block in the virtual machine (column 3, lines 19-35);

formatting the instruction block to include:

an identifier area for storing an identifier that specifies a start position of the instruction block (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9);

a non-branch instruction area for storing non-branch instructions belonging to the corresponding basic block (column 5, lines 36-55);

a branch instruction area for storing at least one branch instruction belonging to the corresponding basic block (column 5, lines 36-55); and

each branch instruction stored in the branch instruction area designating a branch destination using an identifier stored in one of the identifier areas (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9),

wherein the division of the virtual machine instruction sequence into a plurality of separately identifiable instruction blocks having a single branch instruction area reduces the amount of branch destination processing that would otherwise be necessary with a single instruction sequence with branch instructions throughout (Figures 6a, 6b, 7; column 5, line 28 to column 6, line 9).

This is based on the definition of a basic block as offered by United States Patent No. 6,044,222 to Simons et al., in which a basic block is defined as

> A basic block is a contiguous sequence of instructions such that control flow can only enter at the first instruction and leave at the last instruction. Basic blocks have the characteristic that if one of the instructions is ever executed by the program, all the instructions in the basic block will be executed (assuming that no exceptions occur). A basic block typically starts at a branch label and is typically terminated by some sort of branch instruction. *Column 2, lines 15-23*

For a better graphical representation of a basic block please refer to United States Patent No. 5,923,883 to Tanaka et al., Figures 1, 2, 3, 4, 5, 6, 10, 12, 15, 16, 17, 18, 19, 20, 21, 23a, 23b, 24c, 26a, and 28b.

Claim Rejections - 35 USC § 103

11. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

12. Claims 40, 42, and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Adl-Tabatabai in view of United States Patent No. 6,301,652 to Prosser et al., (hereinafter Prosser).

13. As per claim 40, Adl-Tabatabai does not teach wherein the identifier of the instruction block is address related information in the virtual machine instruction sequence.

14. Prosser teaches wherein the identifier of the instruction block is address related information in the virtual machine instruction sequence (column 2, lines 47-60). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the identifiers [labels] of Adl-Tabatabai so that they included address related information much like that of Prosser. One would be motivated to modify the address labeling because it would make more efficient use of memory for execution by performing a cache aligning function. Therefore, one block would lead another block in terms of execution and storage in memory. See column 3, line 62 to column 4, line 2 of Prosser.

15. Regarding claim 42, Adl-Tabatabai a virtual machine instruction at the start position of the basic block being allocated to a specific address in the virtual machine instruction sequence (column 5, lines 36-55), and

a virtual machine instruction at other than the start position of the basic block being allocated to an address other than the specific address (column 5, lines 36-55). This is based on

the definition of a basic block as offered by United States Patent No. 6,044,222 to Simons et al., in which a basic block is defined as

> A basic block is a contiguous sequence of instructions such that control flow can only enter at the first instruction and leave at the last instruction. Basic blocks have the characteristic that if one of the instructions is ever executed by the program, all the instructions in the basic block will be executed (assuming that no exceptions occur). A basic block typically starts at a branch label and is typically terminated by some sort of branch instruction. *Column 2, lines 15-23*

For a better graphical representation of a basic block please refer to United States Patent No. 5,923,883 to Tanaka et al., Figures 1, 2, 3, 4, 5, 6, 10, 12, 15, 16, 17, 18, 19, 20, 21, 23a, 23b, 24c, 26a, and 28b.

16. Adl-Tabatabai does not teach wherein whether each virtual machine instruction is positioned at a start position of the basic block is indicated by an address in the virtual machine instruction sequence to which the virtual machine instruction is allocated.

17. Prosser teaches wherein whether each virtual machine instruction is positioned at a start position of the basic block is indicated by an address in the virtual machine instruction sequence to which the virtual machine instruction is allocated (column 2, lines 47-60). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the identifiers [labels] of Adl-Tabatabai so that they included address related information much like that of Prosser. One would be motivated to modify the address labeling because it would make more efficient use of memory for execution by performing a cache aligning function. Therefore, one block would lead another block in terms of execution and storage in memory. See column 3, line 62 to column 4, line 2 of Prosser.

18. Regarding claim 44, Adl-Tabatabai teaches the basic blocks (Figures 4 [block 430], 6a, 6b, 7; column 5, lines 28-55).

19. Adl-Tabatabai does not teach identification tags, each designates an address related information of the virtual machine instruction at a start position of the basic block; attachment of the tag indicating if the virtual machine instruction corresponding to the identification tag is positioned at the start position of the basic block.

20. Prosser teaches identification tags, each designates an address related information of the virtual machine instruction at a start position of the basic block; attachment of the tag indicating if the virtual machine instruction corresponding to the identification tag is positioned at the start position of the basic block (column 2, lines 47-60). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the identification tags [labels] of Adl-Tabatabai so that they included address related information much like that of Prosser. One would be motivated to modify the address labeling because it would make more efficient use of memory for execution by performing a cache aligning function. Therefore, one block would lead another block in terms of execution and storage in memory. See column 3, line 62 to column 4, line 2 of Prosser.

21. Claims 41 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Adl-Tabatabai in view of Prosser as applied to claim 40 above, and further in view of United States Patent No. 6,151,618 to Wahbe et al., (hereinafter Wahbe).

22. Regarding claim 41, Adl-Tabatabai and Prosser do not teach wherein the address related information is one of absolute address, relative address, and offset address.

23. Wahbe teaches wherein the address related information is one of absolute address, relative address, and offset address (Figures 4 [block 430], 6 [blocks 628, 632, 635], 8 [blocks 811]; column 12, lines 14-30). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the address information of Wahbe in the combined system of Adl-Tabatabai and Prosser. One would be motivated to include the address information because it would make more efficient use of memory for execution by performing memory optimization.

24. Regarding claim 43, Adl-Tabatabai teaches an operation specifying unit for specifying an operation to be executed by the virtual machine (Figure 3 [block 320], 7 [block 705]; column 4, lines 26-43).

25. Adl-Tabatabai does not teach an identifying unit for storing identification information which indicates if the virtual machine instruction is positioned at a start position of the basic block.

26. Wahbe teaches an identifying unit for storing identification information which indicates if the virtual machine instruction is positioned at a start position of the basic block (Figures 4 [block 430], 6 [blocks 625, 628], 8 [blocks 811]; column 12, lines 14-30). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the identifying unit [stack pointer] of Wahbe in the combined system of Adl-Tabatabai and Prosser. One would be motivated to include the identifying unit because it would make more efficient use of memory for execution by performing memory optimization.

Conclusion

27. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

28. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

29. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Christian La Forgia whose telephone number is (703) 305-7704. The examiner can normally be reached on Monday thru Thursday 7-5.

30. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on (703) 305-9648. The fax phone number for the organization where this application or proceeding is assigned is (703) 746-7240.

31. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Christian LaForgia
Patent Examiner
Art Unit 2131
clf

GREGORY MORSE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100